

CUPRINS

11. Arhitectura internă a sistemului <i>Oracle</i>	2
11.1. Nivelul fișiere	2
11.2. Arhitectura proceselor	3
11.2.1. Procese <i>user</i>	3
11.2.2. Procese <i>Oracle</i>	4
11.3. Arhitectura memoriei.....	9
11.3.1. Zona globală sistem	9
11.3.2. Zona globală program	11
Bibliografie	14

11. Arhitectura internă a sistemului *Oracle*

- Din punct de vedere al arhitecturii interne, sistemul *Oracle* este structurat pe trei niveluri:
 - nivelul procese
 - corespunde diverselor procese de sistem care asigură gestiunea datelor;
 - nivelul memorie
 - este compus dintr-o mulțime de zone tampon alocate pentru a stoca date și anumite informații de control;
 - nivelul fișiere
 - corespunde structurii bazei de date și modului în care sunt stocate datele.

11.1. Nivelul fișiere

- Server-ul *Oracle* poate folosi fișiere care nu fac parte din baza de date.
 - Acestea permit configurarea instanței, autentificarea utilizatorilor și recuperarea bazei de date.
- Fișierele de parole (*password file*):
 - sunt fișiere binare folosite pentru autentificarea utilizatorilor bazei de date;
 - sunt localizate în directorul *DATABASE*, iar numele lor depinde de identificatorul *SID* al instanței bazei de date (*psdSID.ora*);
 - pot fi folosite atât pentru conexiunile locale la o bază de date, cât și pentru conexiunile la distanță;
 - în mod automat, sunt ascunse (*hidden*);
 - pot deveni vizibile prin intermediul comenzielor sau utilitarelor specifice fiecărui sistem de operare.
- Fișierul parametrilor de inițializare (*parameter file*)
 - reprezintă principalul mijloc de configurare a sistemului și este utilizat pentru definirea caracteristicilor unei instanțe *Oracle*;
 - conține o colecție de valori ale unor parametri care controlează sau modifică anumite aspecte din modul de funcționare al bazei de date sau al instanței;
 - în versiunile mai vechi, sistemul stoca valorile parametrilor de inițializare doar într-un fișier de tip text (*PFILE*); începând cu versiunea *Oracle9i*, acestea pot fi menținute în fișierul de parametri *server* (*SPFILE*), care este construit prin

comanda *CREATE SPFILE*, pe baza fișierului *PFILE*; sistemul asigură o interfață pentru vizualizarea și modificarea parametrilor de inițializare.

- Fișierele *archived redo log*:
 - sunt copii *offline* ale fișierelor de reluare, folosite pentru recuperarea bazei de date în cazul defectiunilor *hardware*.
- Fișierele istorice (*trace file* și *alert file*):
 - conțin toate mesajele, erorile și evenimentele importante;
 - dacă un proces de tip *server* sau *background* detectează o eroare internă, atunci acesta va scrie informații despre eroarea respectivă în fișierul *trace* asociat; în general, numele fișierelor *trace* asociate cu procesele *background* conțin numele proceselor care le-au generat;
 - orice bază de date conține câte un fișier *alert* asociat fiecărei instanțe;
 - în loc să afișeze informațiile pe consolă, sistemul *Oracle* înregistrează cronologic în acest fișier toate mesajele și erorile care apar.

11.2. Arhitectura proceselor

- Pentru a accesa o instanță a unei baze de date *Oracle*, se execută:
 - aplicație sau un utilitar *Oracle*, prin intermediul cărora se lansează comenzi *SQL* asupra bazei de date;
 - un cod *Oracle server*, cu ajutorul căruia sunt interpretate și procesate comenziile *SQL*.
- Procesele:
 - execută aplicațiile și comenziile interne;
 - sunt mecanisme ale sistemului care permit executarea unor operații de calcul sau operații *I/O*;
 - au alocate zone private de memorie individuale;
 - sunt de două tipuri:
 - procese *user*, care execută aplicațiile;
 - procese *Oracle* (procese *server* și procese *background*), care asigură gestiunea informațiilor dintr-o bază de date.

11.2.1. Procese *user*

- Sistemul creează un proces *user* pentru a executa codul unei aplicații program sau în urma lansării unui utilitar *Oracle*.

- Procesul se execută pe stația *client* (stația de pe care utilizatorul s-a conectat la *server-ul Oracle*).
- Procesul începe și se termină odată cu aplicația utilizatorului respectiv.
- Procesul *user* nu interacționează în mod direct cu *server-ul Oracle*.
 - Aceasta generează mesaje printr-un program interfață (*User Program Interface*) care creează o sesiune și pornește un proces *server*.
 - Programul interfață prelucrează cererile, captează și întoarce erorile, execută conversii și decodificări de date, realizează analize sintactice.
-  ♦ Trebuie făcută distincția dintre conexiune și sesiune.
 - Conexiunea este o cale de comunicare între un proces *user* și o instanță *Oracle*. Aceasta este stabilită prin mecanisme de comunicare între procese sau software pentru rețea.
 - Sesiunea este o conectare a unui *user* la o instanță *Oracle* (printr-un proces *user*). Aceasta durează de la momentul conectării până la deconectare.

11.2.2. Procese *Oracle*

- Procesele *Oracle* execută instrucțiunile interne ale *server-ului Oracle*.
 - Sunt invocate de alte procese pentru a îndeplini anumite operații în favoarea acestora.
 - Există două tipuri de procese *Oracle*:
 - procese *server* (*server process*);
 - procese de fundal (*background process*).

Procesul *server*

- Procesul *server*:
 - interacționează cu procesele *user* și comunică în mod direct cu *server-ul Oracle* pentru a transmite cererile acestora;
 - este pornit atunci când procesul *user* inițiază o sesiune;
 - comunică cu *server-ul Oracle* prin *Oracle Program Interface (OPI)*.
- În cazul configurării *server* dedicate, un proces *server* poate gestiona cererile unui singur proces *user*. Sistemul *Oracle* poate fi configurat astfel încât un singur proces *server* să deservească unul sau mai multe procese *user*. O configurație *server* partajată permite mai multor procese *user* să împartă un număr mic de procese *server*, minimizând numărul de procese *server* și maximizând folosirea resurselor sistem.

- Există sisteme în care procesele *user* și procesele *server* sunt separate și sisteme în care acestea pot fi integrate într-un singur proces. Dacă un sistem folosește un *server* partajat sau dacă procesele *user*, respectiv *server* rulează pe stații diferite, atunci procesele *user* și cele *server* trebuie separate. Sistemele *client/server* separă procesele *user* de cele *server* și le execută pe stații diferite.



- ❖ Server-ul *Oracle* este un sistem de administrare a bazelor de date relaționale orientate pe obiecte și constă din:
 - o instanță *Oracle*;
 - o bază de date *Oracle*.

Instanța Oracle

- Constă dintr-o structură de memorie numită *SGA* (*System Global Area*) și procese *background*.
- Pentru a putea accesa baza de date, trebuie pornită o instanță.
- Instanța poate fi asociată unei singure baze de date.
- Atunci când baza de date este pornită, se localizează structura *SGA* și sunt lansate procesele *background* (*DBWRn*, *LGWR*, *CKPT*, *SMON*, *PMON*, *Dnnn*, *ARCn*, *RECO*, *LMS*, *QMNn* etc.).
- Dacă unul dintre aceste procese se termină anormal, atunci instanța se oprește.

Procesele background

- Sunt folosite pentru a îmbunătăți performanțele unui sistem multiprocesor, în prezența mai multor utilizatori.
- Server-ul *Oracle* creează câte un set de procese *background* pentru fiecare instanță.
 - Acestea reunesc funcțiile executate pentru fiecare proces *user*.
 - De asemenea, procesele *background* execută operațiile *I/O* asincrone și monitorizează alte procese *Oracle*.
 - Procesele *background* *DBWR*, *LGWR*, *CKPT*, *SMON* și *PMON* sunt obligatorii pentru funcționarea sistemului *Oracle*.
- Tipuri de procese *background*
 - *Database Writer* (*DBWR*)
 - Scrie blocurile modificate, din zonele de memorie tampon (*database buffer cache*) în fișierele de date. Această operație este numită *checkpoint* și determină punctul din care începe recuperarea instanței.

- Menține numărul necesar de *buffer*-e libere din *database buffer cache*. Atunci când o zonă tampon din *database buffer cache* se modifică, aceasta este marcată *dirty*. Procesul *DBWR* scrie pe disc zonele tampon *dirty* dacă:
 - un proces *server* are nevoie și nu găsește un *buffer* disponibil;
 - numărul de *buffer*-e *dirty* depășește valoarea admisă;
 - se realizează un *checkpoint* al bazei de date;
 - se elimină sau se trunchiază un tabel;
 - un spațiu tabel devine *offline* sau *read only* etc.
- Nu trebuie să scrie blocurile modificate de fiecare dată când o tranzacție devine permanentă (*commit*). De obicei, *DBWR* se activează atunci când mai multe date trebuie incluse în *SGA* și sunt prea puține zone tampon libere. Datele care nu au fost folosite recent sunt scrise primele în fișierele de date.
- Deși un singur proces *database writer* (*DBWR0*) este suficient pentru majoritatea sistemelor, se pot configura procese adiționale (de la *DBWR1* până la *DBWR9*) cu scopul de a obține performanțe mărite la scriere. Parametrul *DB_WRITER_PROCESSES* specifică numărul de procese *DBWRn*.

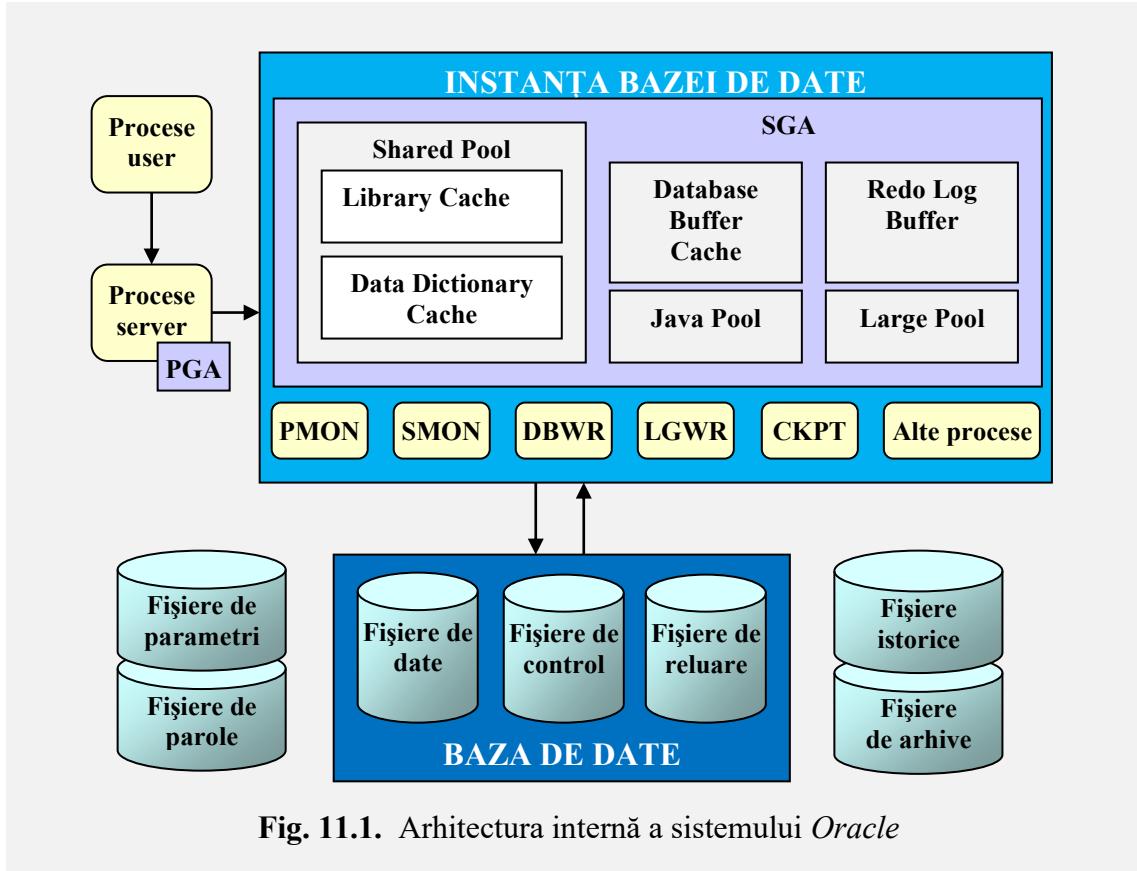


Fig. 11.1. Arhitectura internă a sistemului *Oracle*

- *Log Writer (LGWR)*

- Scrie zonelor tampon de reluare (*redo log buffer*) în fișierele de reluare (*redo log file*):
 - atunci când o tranzacție este permanentizată;
 - înainte ca procesul *DBWR* să scrie blocurile modificate din zonele de memorie tampon în fișierele de date;
 - la fiecare 3 secunde etc.
- Zonele tampon de reluare se găsesc în memoria *SGA* și prin urmare, acest proces înregistrează pe disc modificările care au loc în memoria partajată. Dacă baza de date are fișiere de reluare multiplexate, atunci *LGWR* scrie zonele tampon de reluare într-un grup de astfel de fișiere.

- *Checkpoint (CKPT)*

- Anunță procesul *DBWR* că va avea loc o operație *checkpoint* și modifică *header*-ele fișierelor de date și de control, pentru a înregistra detalii despre aceasta.
- Deoarece procesul *DBWR* scrie blocurile modificate în fișierele de date conform algoritmului *LRU* (*Least Recently Used*), un bloc de date care este folosit frecvent poate să nu fie scris niciodată pe disc dacă nu se realizează o operație *checkpoint*.
- Procesele *CKPT* pot iniția operații *checkpoint* din următoarele motive:
 - pentru a asigura că blocurile de date modificate din memorie sunt scrise pe disc în mod regulat, astfel încât datele să nu se piardă în cazul întreruperilor la nivelul sistemului sau al bazei de date;
 - pentru a reduce timpul necesar recuperării instanței (în acest caz, este necesară doar procesarea fișierelor de reluare);
 - pentru a asigura că toate datele permanentizate sunt scrise în fișierele de date în timpul procesului de închidere al bazei de date.

- *System Monitor (SMON)*

- Inițiază procesul de recuperare a datelor la pornirea unei instanțe care a eșuat anterior.
- Procesul de recuperare al unei instanțe se realizează în mai mulți pași.
 - 1) Sunt recuperate datele care nu au fost înregistrate în fișierele de date, dar care au fost înregistrate în fișierele de reluare. În timpul acestei operații, procesul *SMON* citește fișierele de reluare și modifică blocurilor de date în

conformitate cu acestea. Deoarece tranzacțiile permanentizate au fost scrise în fișierele de reluare, procesul recuperează complet aceste tranzacții.

2) Se deschide baza de date, astfel încât utilizatorii să se poată conecta. Toate datele care nu sunt blocate de tranzacțiile nerecuperate devin imediat disponibile.

3) Se anulează (*rollback*) tranzacțiile nepermanentizate.

- De asemenea, procesul *SMON* îndeplinește și funcții de întreținere a spațiului. Acesta reunește zonele adiacente de spații libere din fișierele de date și asigură eliberarea segmentelor temporare care nu mai sunt utilizate.

- *Dispatcher (Dnnn)*

- Este un proces *background* optional, prezent atunci când este folosită o configurație *server* partajată.
- Pentru fiecare protocol de comunicare este creat cel puțin un proces dispecer.
- Fiecare proces dispecer este responsabil cu direcționarea cererilor venite din partea unor procese *user*, cu validarea proceselor *server* și transmiterea răspunsului către procesele *user*.

- *Process Monitor (PMON)*

- Asigură recuperarea proceselor *user*, dacă acestea eșuează. În acest sens, procesul *PMON* este responsabil pentru anularea tranzacției curente, eliberarea memoriei *cache* și a resurselor utilizate de procesul respectiv, rezolvarea blocărilor curente intervenite la nivel de tabel sau linie.
- De asemenea, *PMON* verifică procesele *server* și procesele dispecer pe care le repornește dacă acestea eșuează.

- *Recoverer (RECO)*

- Este folosit pentru a rezolva problema tranzacțiilor distribuite dintr-o rețea și căderile sistemului unei baze de date distribuite.
- Procesul *RECO* se conectează automat la bazele de date distante ale căror tranzacții distribuite se derulează anormal și execută operații de permanentizare sau de anulare pe porțiuni locale din tranzacțiile distribuite în așteptare.
- Dacă conexiunea eșuează, atunci procesul încearcă, la anumite intervale de timp, să stabilească o nouă legătură cu bazele de date implicate.

- *Archiver (ARCn)*

- Produce copia arhivată a fișierelor de reluare, înregistrând toate modificările făcute în baza de date.

- Deși pentru cele mai multe sisteme este suficient un singur proces *archiver* (*ARC0*), se pot specifica mai multe astfel de procese, prin intermediul parametrului de initializare *LOG_ARCHIVE_MAX_PROCESSES*.
- *Lock Manager Server (LMS)*
 - Gestionea inter-blocările dintre instanțele bazei de date în *Oracle Real Application Clusters*.
- *Queue Monitor (QMNn)*
 - Este un proces *background* optional care monitorizează mesajele din lista creată prin *Oracle Advanced Queuing*.

11.3. Arhitectura memoriei

- Toate structurile de memorie se găsesc în memoria centrală. Sistemul creează și folosește structurile de memorie pentru a depozita codul programelor executate, datele necesare în timpul execuției acestora, datele folosite în comun de mai multe procese *Oracle*, informațiile referitoare la sesiunile curente etc.
- Din punct de vedere structural, memoria este compusă din:
 - zona globală sistem (*SGA - System Global Area*);
 - zona globală program (*PGA - Program Global Area*).

11.3.1. Zona globală sistem

- Este un grup de structuri partajate de memorie care conțin date și informații de control relative la o instanță.
- Este alocată atunci când se pornește instanța și este eliberată în momentul opririi acesteia.
 - Fiecare instanță are propria sa *SGA*.
 - Datele conținute în *SGA* sunt folosite în comun de către utilizatorii conectați la instanță.
 - O parte din *SGA* este folosită pentru stocarea informațiilor despre starea bazei de date și a instanței. Informațiile sunt accesate de către procesele *background*. Această zonă este numită *SGA fixă* și nu poate conține date ale utilizatorilor.
- Trebuie să fie dimensionată astfel încât să permită stocarea cât mai multor date în memorie și să reducă numărul de operații *I/O*.

- Începând cu versiunea *Oracle9i*, sistemul poate configura în mod dinamic zona *SGA* în timp ce instanța este pornită (*SGA* dinamic).
- Informațiile conținute în *SGA* sunt repartizate în diferite zone (care sunt alocate la pornirea instanței):
 - *database buffer cache*,
 - *redo log buffer*,
 - *shared pool*,
 - *large pool*,
 - *Java pool*.

Database buffer cache

- Este o mulțime de zone tampon care stochează blocurile de date (modificate sau nu) folosite cel mai recent.
- Aceste zone sunt organizate în două structuri:
 - listă de blocuri scrise (*write list*)
 - este formată din *buffer*-ele *dirty*, care conțin datele modificate, dar nu scrise pe disc;
 - listă de blocuri care au fost accesate recent (*least recently used list*).
 - este formată din *buffer*-e libere, *buffer*-e din memoria *cache* și *buffer*-e *dirty* care nu au fost încă mutate în lista de blocuri scrise.
- Atunci când este executată o cerere, procesul *server* caută blocurile de care are nevoie în *database buffer cache*. Dacă nu găsește un bloc, procesul *server* îl citește din fișierul de date și plasează o copie a acestuia în *database buffer cache*. Întrucât datele cel mai frecvent folosite sunt păstrate în memoria *cache*, se micșorează numărul de operații *I/O* și se îmbunătățesc performanțele bazei de date.

***Redo log buffer* (zona tampon de reluare)**

- Este un *buffer* circular care conține informațiile referitoare la modificările blocurilor de date ale bazei.
- Aceste informații pot fi necesare atât pentru recuperarea bazei de date, cât și pentru întoarcerea la forma anterioară a modificărilor realizate prin operații de tip *LMD* sau *LLD*.
- Conținutul *buffer*-ului este scris de procesul *LGWR* în fișierul de reluare.

Shared pool

- Este o porțiune din *SGA* formată din trei zone partajate de memorie:
 - *library cache*
 - conține informații despre cele mai recente comenzi *SQL* (text, versiune compilată, plan de execuție) și unități de program *PL/SQL*;
 - o cerere executată de mai multe ori poate utiliza versiunea compilată și planul de execuție deja existent de la rularea anterioară;
 - *data dictionary cache*
 - conține cele mai recente informații accesate din dicționarul datelor;
 - această zonă de memorie poartă și denumirea de *row cache*, deoarece datele sunt stocate sub formă de linii (nu sub formă de *buffer*-e care conțin blocuri întregi de date);
 - *buffer*-e pentru mesaje relative la execuțiile paralele și la structurile de control.

Large pool

- Este o zonă opțională *SGA*, configurată doar pentru *server*-ele partajate.

Java pool

- Este o zonă de memorie opțională, necesară atunci când este instalat și utilizat limbajul *Java*.

11.3.2. Zona globală program

- *PGA* este o zonă de memorie care conține date și informații de control relative la un singur proces *Oracle*.
 - ❖ Spre deosebire de *SGA*, care este utilizată de mai multe proceze (zonă de memorie partajată), *PGA* este folosită numai de unul singur (zonă de memorie nepartajată).
 - ❖ Zona globală program este alocată la crearea procesului și eliberată la terminarea acestuia.
- Conținutul zonei de memorie *PGA* depinde de modul de configurare a sistemului (*server* dedicat sau partajat), singurul care poate accesa această zonă.



! ❖ Spre deosebire de *SGA*, care este utilizată de mai multe proceze (zonă de memorie partajată), *PGA* este folosită numai de unul singur (zonă de memorie nepartajată).

❖ Zona globală program este alocată la crearea procesului și eliberată la terminarea acestuia.

- În general, zona globală program este formată din
 - zonă privată *SQL*;
 - conține date (de exemplu, informații de legătură) și structuri de memorie necesare rulării comenziilor;
 - pentru fiecare comandă *SQL* inițiată de o sesiune se asociază o zonă privată *SQL*;
 - în cazul *server*-elor partajate, zona privată *SQL* este menținută în *SGA*;
 - administrarea zonelor private *SQL* este responsabilitatea procesului *user*;
 - numărul de zone private *SQL* pe care le poate aloca procesul *user* este limitat de valoarea parametrului de initializare *OPEN_CURSORS*;
 - zonă de memorie alocată sesiunii;
 - zone de lucru *SQL*.

Scenariu

- Următoarea succesiune de pași ilustrează o configurare *Oracle* în care procesul *user* și procesul *server* asociat rulează pe mașini diferite.
 - 1) Se pornește o instanță pe stația care găzduiește sistemul *Oracle* (cunoscută sub denumirea de gazdă sau *server* de baze de date).
 - 2) O stație de lucru *client* rulează o aplicație care determină crearea unui proces *user*. Aplicația *client* încearcă să stabilească o conexiune cu *server*-ul de baze de date.
 - 3) *Server*-ul detectează cererea de conectare a aplicației și creează un proces *server* dedicat procesului *user*. Procesului *server* i se alocă o zonă de memorie *PGA*.
 - 4) Utilizatorul execută o comandă *SQL* și apoi instrucțiunea *COMMIT* pentru a permanentiza schimbările și a încheia tranzacția.
 - 5) Procesul *server* primește comanda și verifică dacă în *shared pool (library cache)* există o zonă partajată *SQL* care conține comenzi similare. Dacă este găsită o astfel de zonă, procesul *server* verifică privilegiile de acces ale utilizatorului la datele interogate și folosește zona respectivă pentru procesarea comenzi. În caz contrar, pentru comandă este alocată o nouă zonă partajată *SQL*, unde aceasta va putea fi analizată și procesată.
 - 6) Procesul *server* extrage datelor necesare cereri. Dacă datele necesare au fost recent folosite, atunci procesul *server* le va găsi și extrage din *database buffer cache*. În caz contrar, datele vor fi extrase din fișierele de date și o copie a acestora va fi depusă în *database buffer cache*.

- 7) Procesul *server* modifică datele din *database buffer cache*. Informațiile referitoare la modificările blocurilor de date sunt înregistrate în *redo log buffer*. Deoarece tranzacția a fost permanentizată, procesul *LGWR* o înregistrează imediat într-un fișier de reluare. La un moment dat procesul *CKPT* anunță procesul *DBWR* că va avea loc o operație *checkpoint* și modifică *header*-ele fișierelor de date și de control, pentru a înregistra detalii despre aceasta. *DBWR* scrie blocurile de date modificate din *database buffer cache* în fișierele de date.
 - 8) Dacă tranzacția s-a derulat cu succes, procesul *server* trimite un mesaj corespunzător către aplicație. Altfel, este transmis un mesaj de eroare.
- Pe întregul parcurs al procedurii, rulează și alte procese *background*, care pot interveni în condiții exceptionale. În plus, *server*-ul bazei de date administrează și alte tranzacții, prevenind inter-blocarea celor care necesită aceleași date.

Bibliografie

1. Connolly T.M., Begg C.E., Database Systems: *A Practical Approach to Design, Implementation and Management*, 5th edition, Pearson Education, 2005
2. Dollinger R., Andron L., *Baze de date și gestiunea tranzacțiilor*, Editura Albastră, Cluj-Napoca, 2004
3. Oracle and/or its affiliates, *Oracle Database Concepts*, 1993, 2025
4. Oracle and/or its affiliates, *Oracle Database Performance Tuning Guide*, 2013, 2025
5. Oracle and/or its affiliates, *Oracle Database SQL Language Reference*, 1996, 2025
6. Oracle and/or its affiliates, *Oracle Database PL/SQL Language Reference*, 1996, 2025
7. Oracle and/or its affiliates, *Oracle Database Administrator's Guide*, 2001, 2023
8. Oracle and/or its affiliates, *Pro*C/C++ Programmer's Guide*, 1996, 2019
9. Oracle University, *Oracle Database: PL/SQL Fundamentals, Student Guide*, 2009, 2025
10. Popescu I., Alecu A., Velcescu L., Florea (Mihai) G., *Programare avansată în Oracle9i*, Ed. Tehnică, 2004